

Modulname	Python Programmierung 2		
Lead	Python hat sich als eine der beliebtesten Programmiersprachen etabliert, weil sie die besten Konzepte aus verschiedenen Sprachen vereint und gleichzeitig einfach zu erlernen ist. In diesem Modul knüpfen wir an den vorherigen Kurs an und nehmen Sie mit auf eine spannende Reise, um noch tiefer in die Welt von Python einzutauchen. Sie entdecken fortgeschrittene Eigenschaften und Konstrukte, perfektionieren Ihre Nutzung der Entwicklungsumgebung und entwickeln Ihre Fähigkeiten im Software Engineering weiter – zum Beispiel durch den Einsatz des Test-Driven Development-Ansatzes. Werden Sie zum Python-Profi und bringen Sie Ihre Projekte auf das nächste Level!		
Zielgruppe	Studierende der Höheren Fachschule in den Fachrichtungen Automation, Elektrotechnik, Informatik und Erneuerbare Energie		
Inhalt	<ul style="list-style-type: none"> <li>• Klassen und Klassen ableiten</li> <li>• Methoden und Operator overloading</li> <li>• API Doc für eigenen Klassen erstellen</li> <li>• Exception-Handling und eigene Exceptions</li> <li>• Multithreading</li> <li>• JSON/XML-Processing</li> <li>• Hardware ansteuern: GPIO und PiPlates,</li> <li>• Template-Mechanism (JINJA2)</li> <li>• REST-Service implementieren (FLASK)</li> <li>• Web-Applikationen entwickeln (Sessions, Bootstrap, Formulare, AJAX)</li> </ul>		
Kompetenzen	A02.50	Informations- und Kommunikationstechnologien (ICT) professionell einsetzen und etablieren	
	A03.20	Neues Wissen mit geeigneten Methoden erschliessen und arbeitsplatznahe Weiterbildung realisieren	
	A03.30	Neue Technologien kritisch reflexiv beurteilen, adaptieren und integrieren	
	A03.40	Die eigenen digitalen Kompetenzen kontinuierlich weiterentwickeln	
	B10.10	Die Architektur der Software bestimmen und die Entwicklung unter Berücksichtigung von Betrieb und Wartung planen und dokumentieren	
	B11.30	Spezifikation in einer geeigneten Programmiersprache umsetzen	
	B11.40	Entwicklungsprojekte aufgrund der Analyseergebnisse und des gewählten Vorgehens planen und leiten	
	B11.50	Mobile und verteilte Applikationen unter Berücksichtigung zeitgemässer Architekturmuster bzw. Referenzarchitekturen implementieren	
	B11.60	Testkonzepte und Testspezifikation erstellen, Tests implementieren und auswerten sowie notwendige Massnahmen umsetzen	
	B11.80	Prinzipien, Methoden und Werkzeuge für die arbeitsteilige Entwicklung und Anwendung von umfangreichen Softwaresystemen zielorientiert bereitstellen und systematisch umsetzen	
Ziele	<p><b>K2 – Verstehen</b></p> <ol style="list-style-type: none"> <li>1. <b>Sie können</b> die Methoden <code>__init__</code> <code>__str__</code> in einer Klasse erläutern und deren Einsatzmöglichkeiten aufzeigen.</li> <li>2. <b>Sie können</b> einen http Request und eine JSON/XML-Response beschreiben.</li> <li>3. <b>Sie können</b> den Unterschied zwischen Modul, Packages und Klassen und Bibliotheken erläutern.</li> <li>4. <b>Sie können</b> erläutern was eine ENUM ist.</li> </ol> <p><b>K3 – Anwenden</b></p> <ol style="list-style-type: none"> <li>5. <b>Sie können</b> eine eigene Klasse designen, implementieren und testen</li> <li>6. <b>Sie können</b> die Sichtbarkeitsstufen von Methoden und Properties richtig nutzen.</li> <li>7. <b>Sie können</b> ENUMS sinnvoll nutzen.</li> <li>8. <b>Sie können</b> komplexere Applikationen in Python schreiben und Konzepte von Sub-Classing, Multithreading, ENUMS und Template-Mechanism (JINJA) verwenden.</li> </ol> <p><b>K4 – Analysieren</b></p> <ol style="list-style-type: none"> <li>9. <b>Sie können</b> API-Dokumentationen analysieren und daraus ableiten, wie externe Pakete in eigene Anwendungen integriert werden können.</li> <li>10. <b>Sie können</b> bestehende Python-Applikationen untersuchen und deren Funktionalität sowie Schwachstellen identifizieren.</li> </ol>		

11. **Sie können** öffentliche REST-Services analysieren und in eigenen Python-Applikationen nutzen.
12. **Sie können** die Architektur einer Python-Anwendung analysieren, um Verbesserungspotenziale zu erkennen.

**K5 – Bewerten**

13. **Sie können** die Vor- und Nachteile von service-orientierter gegenüber funktionaler Programmierung in Python bewerten.
14. **Sie können** verschiedene Ansätze für die Steuerung von Aktoren auf einem Raspberry Pi kritisch vergleichen und Empfehlungen aussprechen.
15. **Sie können** die Effizienz und Sicherheit verschiedener Methoden für den Zugriff auf Raspberry Pi-Systeme beurteilen.
16. **Sie können** den Einsatz von Sense-HAT-Modulen und PiPlates in einem Projekt evaluieren und deren Nutzen für spezifische Anwendungen bewerten.

Voraussetzungen	Erfolgreicher Abschluss von Modul <b>Python Programmierung 1</b>
Lehrmittel	